

A Study on Feature Selection for the Detection of WannaCry, Ryuk, and CryptoLocker Ransomwares

Gabriel O. Souza¹, Silvio E. Quincozes¹, Juliano F. Kazienko²
Vagner E. Quincozes³ e Nicolás Naves R. Faria¹

¹ FACOM – Universidade Federal de Uberlândia (UFU), Monte Carmelo, Brasil

²CTISM – Universidade Federal de Santa Maria (UFSM), Santa Maria, Brasil

³Universidade Federal do Pampa (UNIPAMPA), Alegrete, Brasil

{gabrieloliveirasouza620, sequincozes, nicolasnaves}@ufu.br,
kazienko@redes.ufsm.br, vagnerquincozes.aluno@unipampa.edu.br

Abstract. Ransomware is a malicious software that encrypts files and demands a ransom payment. Due to their late detection, they pose a significant threat to individuals and organizations, resulting in substantial financial losses and data breaches. This paper identifies relevant features for the early detection of three well-known types of ransomware: WannaCry, Ryuk, and CryptoLocker. We use the Cuckoo environment to carry out experiments in which each ransomware activity is logged. Afterward, we create and make available specialized datasets for each studied ransomware attack containing normal and attack samples. Finally, we apply feature ranking methods to identify the 20 most representative features for detecting each ransomware family from our datasets, providing a valuable contribution towards improving ransomware detection mechanisms.

Resumo. Ransomware é um malware que sequestra dados e devido a sua detecção frequentemente tardia apresentam uma ameaça significativa para indivíduos e organizações, causando perdas financeiras substanciais e violações de dados. O objetivo deste trabalho consiste em identificar as features relevantes para detecção de três tipos conhecidos de ransomwares: WannaCry, Ryuk e CryptoLocker. Para tanto, é utilizado o ambiente Cuckoo Sandbox no qual a atividade de tais ransomwares são registradas. Com isso, são criados datasets contendo amostras normais e de ataque para cada tipo de ransomware estudado. Por fim, aplicou-se métodos de ranqueamento de features nesses datasets para a identificação das 20 features mais relevantes para a detecção dos ransomwares estudados, fornecendo uma contribuição valiosa para a melhoria dos mecanismos de detecção de ransomware.

1. Introdução

The COVID-19 pandemic severely impacted the way people work. Research conducted in over 200 companies shows that 81% of them adopted the remote work model [Curran 2020]. As a result, the occurrence of attacks through malicious software (*malwares*) increased by more than 75% in the first half of 2022 [Micro 2022][Razaulla et al. 2023]. Among the most devastating *malwares*, ransomware stands out. In the current scenario, this type of *malware* emerges as one of

the main threats, with a devastating impact [Abbasi et al. 2022][Brewer 2016]. These *malwares* hijack and encrypt valuable data, demanding payment of a ransom for its release [Urooj et al. 2022]. Some types of *ransomwares* have gained prominence due to their popularity, occurrence rate, and primarily financial damage caused, such as *WannaCry*, *Ryuk*, and *CryptoLocker* [Beaman et al. 2021].

Despite the common incidence of *ransomwares* in corporate environments today, most attacked companies are still susceptible to losing their data or being forced to pay the ransom. Even the loss of control or deactivation of devices can have harmful effects, especially in industrial control systems, or *Industrial Control Systems* (ICSs), and typical devices in these systems, such as *Human Machine Interfaces* (HMIs) and *Programmable Logic Controllers* (PLCs) [Micro 2020]. A report by the Russian cybersecurity company *Kaspersky* reveals that about 88% of companies that have already suffered a *ransomware* attack admit they would pay the ransom if attacked again [Kaspersky 2021]. This is mainly due to the lack of adequate solutions to mitigate *ransomwares*. There have been efforts in the literature to detect *ransomwares* through the processing of features that represent the operational pattern of these *malwares*. However, there is no consensus on which features are most effective in detecting *ransomwares*.

In the literature, there are authors who focus on studying the behavior of *ransomwares* such as *WannaCry* [Chen and Bridges 2017][Chen et al. 2019], *Ryuk* [Masid et al. 2022], and *CryptoLocker* [Abbasi et al. 2020][Abbasi et al. 2022]. These works employ tools such as *Cuckoo* [Oktavianto and Muhandianto 2013] for feature extraction and, in some cases, machine learning algorithms with the intention of selecting representative features. However, these works present as their main limitations the modeling of legitimate user behaviors, which is used as a contrast to the operations performed by *ransomwares*. In particular, such modeling involves little diversity of operations and is potentially biased, as these operations (*e.g.*, online ticket purchases) are not similar to those performed by a *ransomware*, which may result in non-representative normal samples negatively impacting the representativeness of the selected features.

This study aims to identify important characteristics for detecting three types of ransomware: *WannaCry*, *Ryuk*, and *CryptoLocker*. Using the *Cuckoo Sandbox* tool, the behaviors of normal files and the studied ransomwares were captured, generating three corresponding datasets. The results obtained through the *Releaf* feature selection algorithm indicate the 20 most relevant *features* for detecting the three types of *ransomwares* studied. The study of these *ransomwares* is crucial for understanding the cyber threats most prevalent in the current scenario. The motivation for studying these *malwares* lies in the need to understand how they work, how they spread, and how they can be detected and prevented efficiently. Each *ransomware* has its own characteristics and attack techniques [IBM 2022].

2. Theoretical Background

This section introduces the concept of *ransomware* and its categories, as well as the types studied in this work. Additionally, feature selection and its importance in the detection process of this *malware* are discussed. By studying the *ransomwares* in question, it is possible to understand the techniques used by attackers, such as exploiting system vulnerabilities, using social engineering, and employing encryption techniques. Furthermore, it

is possible to learn about the Windows APIs and other technologies used by *ransomwares*, which will be mentioned later.

2.1. Ransomware

Ransomware is a type of *malware* that is capable of preventing access to personal data unless a ransom is paid. Typically, the ransom for the data is paid through cryptocurrencies, which at the same time creates a major dilemma regarding the tracking of the transaction requested by cybercriminals [Razaulla et al. 2023].

Ransomwares can be divided into three categories: *Scareware*, which uses advertisements to manipulate users into downloading a specific type of software containing malicious code snippets; *Locker*, which aims to block any functionality of the computer (e.g., screen, keyboard), preventing the user from navigating or using the machine; and *Crypto*, which encrypts the user's disk files but does not interfere with the basic functions of the computer. In general, all three categories use phishing emails as the main form of dissemination, aiming to facilitate the contamination of target machines and gradually spreading through the network [Beaman et al. 2021].

In this work, three variations of *ransomwares* are analyzed: *WannaCry*, *CryptoLocker*, and *Ryuk*. The first type is *WannaCry*, which is one of the most popular and traditional *ransomwares* in the *crypto* category, where its main propagation method is through *worms* [Beaman et al. 2021]. *Worms* are a category of *malware* capable of copying themselves from machine to machine, usually exploiting some type of security flaw in a software or operating system, without requiring user interaction to operate. Another type, *Ryuk*, is also in the *crypto* category. Its main method of propagation is through phishing emails [Beaman et al. 2021]. The emails sent to the user usually have a forged origin. The *ransomware* attack begins, for example, when the target downloads an attached file to their machine and executes it. According to a report by Trend Micro [Micro 2020], *Ryuk* was responsible for most of the attacks on ICSs in 2020. Lastly, the *ransomware* *CryptoLocker* encrypts the user's files and demands a ransom in exchange for the decryption key. It belongs to the *crypto* category. Its dissemination usually occurs through phishing emails, which contain malicious attachments or links to forged websites. This *ransomware* is listed as the seventh most frequent *malware* by the *Center for Internet Security* in 2022 [CIS 2022].

2.2. Feature Selection

Feature selection consists of choosing the characteristics, or *features*, that best describe a particular situation, such as a cyberattack. Specifically, the characteristics are intended to identify attacks caused by the *ransomwares* studied in this work. The choice of *features* significantly impacts the accuracy rate, among other metrics, in a detection process. Therefore, a thorough selection effort is crucial [Quincozes et al. 2018].

Relief is an algorithm that uses a statistical method to identify the most relevant *features* in datasets. It is efficient in terms of execution time and tends to select a small subset of statistically relevant characteristics. To achieve this, the algorithm uses a sample-based approach, randomly selecting an instance and then finding the closest instance that belongs to a different class. It then updates the weights of all attributes based on the differences between these two instances. A comparison with other attribute selection algorithms highlights the advantages of Relief in terms of learning time and accuracy of the

learned concept, suggesting the practicality of Relief [Kira and Rendell 1992]. Below, in Algorithm 1, the pseudocode of the Relief algorithm (based on [Megchelenbrink 2010]) is shown.

Algoritmo 1 Feature Selection Based on the Relief Algorithm	
1: Input: Dataset D , number of features nf	
2: Output: Set of selected features C	
3: $C \leftarrow \emptyset$	▷ Initialize the set of selected features with an empty value
4: $P \leftarrow \emptyset$	▷ Initialize the weight vector P
5: for $f \leftarrow 1$ to nf do	
6: $P_f \leftarrow 0$	▷ Initialize the weight of feature f
7: end for	
8: for $i \leftarrow 1$ to $ D $ do	
9: $x_i \leftarrow \text{random}(D)$	▷ Randomly select (x) an instance i from D
10: $amp \leftarrow \text{nearestSameClass}(x_i)$	▷ Instance with the nearest same class
11: for $f \leftarrow 1$ to nf do	
12: $P_f \leftarrow P_f - \frac{1}{ D }(x_{i,f} - x_{amp,f})$	▷ Update the weight of f based on amp
13: $emp \leftarrow \text{nearestDifferentClass}(x_i)$	▷ Instance with the nearest different class
14: $P_f \leftarrow P_f + \frac{1}{ D }(x_{i,f} - x_{emp,f})$	▷ Update the weight of f based on emp
15: end for	
16: end for	
17: for $f \leftarrow 1$ to amp do	
18: $i \leftarrow \arg \max_{i \in \{1, \dots, nf\} \setminus C} P_i$	▷ Select the feature with the highest weight
19: $C \leftarrow C \cup f$	▷ Add the selected feature to the set
20: end for	
21: return C	

3. Related Works

In the literature, there are several works related to the study of *ransomwares*, but there are still gaps regarding the analysis of *features* for their detection.

In [Razaulla et al. 2023], a comprehensive overview of the state of the art in the context of *malwares* of the *ransomware* type is presented, including its evolution, taxonomy, detection, and infection vectors. Thus, this work provides an updated understanding of the cybersecurity field of *ransomware* and encourages the fight against this constantly evolving threat. However, aspects related to the *features* that may be relevant for detecting this *malware* are not addressed.

Another survey study is conducted by [Urooj et al. 2022], which aims to contribute to the comprehensive understanding of the evolution and classification of *ransomware*, as well as highlight the main ongoing research in this area. Through a detailed analysis of more than 150 references, it was observed that most studies (72.8%) focused on *ransomware* detection, with a significant proportion (70%) applying Machine Learning techniques. However, there is a gap concerning the lack of studies on more recent *ransomwares*.

In [Chen and Bridges 2017], an automated behavioral analysis of the *WannaCry ransomware* is presented. The methodology adopted by the authors consists of generating *features* using the *Cuckoo* tool [Oktavianto and Muhandianto 2013] and feature selection with the machine learning algorithm *Term Frequency–Inverse Document Frequency* (TF-IDF). In particular, executable binaries of *WannaCry* were used, and Python scripts were

developed to simulate normal behavior. The main limitations of this work arise from the modeling of legitimate behavior, which involves little diversity of operations and is potentially biased since these operations (e.g., online ticket purchases) are not similar to those performed by a *ransomware*.

In [Chen et al. 2019], the authors present an automated tool for pattern extraction and early detection of *ransomwares*, including *WannaCry*. For the feature generation step, the authors execute the *Cuckoo* tool [Oktavianto and Muhandianto 2013] following the same methodology used in [Chen and Bridges 2017]. Then, three machine learning algorithms were employed to discriminate the most relevant *features*: TF-IDF, *Fisher's Linear Discriminant Analysis* (LDA), and *Extremely Randomized Trees* (ET). However, although the main contribution is the selection of *features*, the generation of patterns representing normal behavior is limited to simple and predictable operations, similar to those in [Chen and Bridges 2017].

In [Gera et al. 2021], the authors propose a hybrid approach to identify and mitigate *ransomware* for the Android platform. A new feature selection algorithm is employed to extract the dominant feature set. Experimental results show that the proposed model can identify *ransomware* with greater accuracy. However, despite the study involving a widely used mobile platform, the authors do not evaluate the types of *ransomwares* studied in this work.

In the works [Abbasi et al. 2020] and [Abbasi et al. 2022], the authors present a feature selection method based on a *wrapper* approach to detect and classify *ransomwares* — including *CryptoLocker* — based on their behavior. A group-oriented strategy and *Particle Swarm Optimization* (PSO) are used to select an appropriate number of *features* from each group, based on their relevance. However, the authors do not specify which benign binaries were used in the research. As a result, it is not possible to determine the degree of similarity of the actions performed by the legitimate binary in relation to the malicious binary.

A methodology for analyzing *malwares* in general is proposed in [Masid et al. 2022]. As a proof of concept, the authors apply this methodology in the form of a case study on the *Ryuk ransomware*. To do so, information beyond its behavior is captured through a hybrid analysis. However, in contrast to the proposal of this work, the authors do not indicate which *features* are most relevant for detecting *Ryuk*.

Thus, although the current literature includes the works discussed in this section, there are still gaps that need to be addressed. For example, one of the main limitations in the literature is the modeling of legitimate user behaviors, which is used as a contrast to the operations performed by *ransomwares*. As a result, there is little diversity of legitimate operations being studied in parallel with the behavior of *ransomwares*, which may hinder the efficient identification of the most relevant information for detecting this type of *malware*.

4. Materials and Methods

This section presents the materials and methods adopted in this work. In Section 4.1, the studied scenario is presented. Then, in Section 4.2, the steps of the study based on the previously presented scenario are outlined.

4.1. Scenario

In order to reproduce an environment with both legitimate and malicious operations, a virtualized environment was configured using the *Cuckoo Sandbox* tool [Oktavianto and Muhandianto 2013]. This tool is a free open-source *software* that allows automated analysis of binaries (*malwares*). The choice of this tool was based on (i) its offering of a variety of options for malware analysis, surpassing *sandboxes* such as Drakvuf, and (ii) its ability to be enhanced with additional tools, such as YARA and Suricata [Ilić et al. 2022]. From the environment configured in Cuckoo Sandbox, 35 binaries of legitimate applications (*e.g.*, Ruffus, Chrome, Firefox, Putty, Tetris, etc.) and 3 *ransomwares*, namely: *WannaCry*, *Ryuk*, and *CryptoLocker* were executed. The *ransomwares* were run three times on the *Cuckoo Sandbox*. Each binary execution was set to a runtime of ten minutes.

To facilitate the reproducibility of the results presented in Section 5, the benign and malicious binaries, *datasets*, and *scripts* used in the experiments are available in a repository created on the *GitHub* platform¹. Figure 1 illustrates the experimental scenario, which involves uploading binaries (*binary.exe*) to the tool and creating virtual machines using the VirtualBox software. The *Cuckoo* tool has an agent, created in Python, that monitors and captures all content and changes made by the binary during its execution. The tool is installed on a Dell Inspiron 15 3000 laptop with Ubuntu 18.04 operating system.

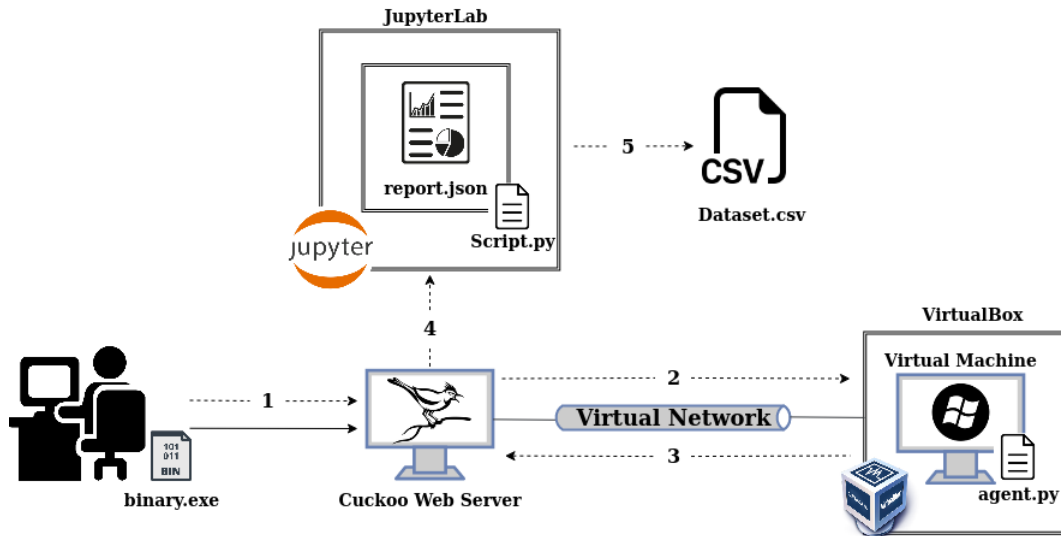


Figura 1. Cuckoo Sandbox - Flow

4.2. Methodology

Once the virtualized environment for executing binaries was built, analyses of these binaries were performed using both static and dynamic approaches. The main difference between these approaches is that static analysis relies solely on the analysis of the source code of the binaries and does not require execution, in contrast to dynamic analysis, which examines the flow of information and function calls of a running

¹Available at: <https://github.com/gabrielolivs/Deteccao-de-Ransomware>

binary. When both techniques are used together, this process is called hybrid analysis [Damodaran et al. 2017][Uppal et al. 2014].

The steps illustrated in the scenario of Figure 1 demonstrate the methodology employed in conducting the experiments. In Step 1, the user inputs the binary to be analyzed, specifying the execution time and allowing internet access for other actions during its execution. Then, in Step 2, the application uses the *Virtual Network* to send and initiate the binary execution on the virtual machine configured for the *Cuckoo Sandbox* to perform its analysis. During the binary execution, a program called `agent.py` runs inside the virtual machine, capturing all actions of the binary.

After the binary execution is completed, in Step 3, the `agent.py` captures all relevant information that occurred during execution and sends it to the application via the *Virtual Network*. The *Cuckoo Sandbox* collects all this information and stores it in a *JavaScript Object Notation* (JSON) file, which is sent in Step 4 to the JupyterLab platform. On this platform, code was implemented to collect and filter the relevant information for ranking the *features*, as described in this paper.

To process the report resulting from the *Cuckoo* analysis in this component, a *Python* script called `script.py` was implemented and is available in the previously mentioned repository. It is responsible for creating a CSV file containing all the filtered features, as shown in Step 5 in the image.

From the file resulting from the process illustrated in Figure 1, 226 *features* were collected for *WannaCry*, 211 for *Ryuk*, and 211 for *CryptoLocker*. These *features* represent information such as system calls and operations performed by each executed binary. Thus, three *datasets* were constructed with malicious samples (*WannaCry*, *Ryuk*, *CryptoLocker*) and legitimate samples (typical user applications). It is important to highlight that each generated *dataset* contains samples from 35 legitimate binaries and 1 malicious binary. Therefore, what differentiates one *dataset* from another are the samples of the malicious binary. In other words, each *dataset* contains samples from only one of the malicious binaries, along with samples from the 35 legitimate binaries, which are the same for each dataset.

Finally, the Relief algorithm² was used to measure the representativeness of each of the generated *features*, thus discarding the less representative ones. This algorithm is detailed in Section 2.2.

5. Results and Discussion

This section presents the results obtained for each *ransomware* studied, along with the corresponding discussions.

5.1. WannaCry Study

From the 226 *features* generated, only 20 have a weight greater than 0.6. Therefore, as preliminary results, these *features* that make up the *Top 20* are presented and discussed. Figure 2 illustrates the weight computed by the Relief algorithm for each of the *features* among the top 20 highest-rated ones.

²Available in the WEKA library: <https://www.cs.waikato.ac.nz/ml/weka/>

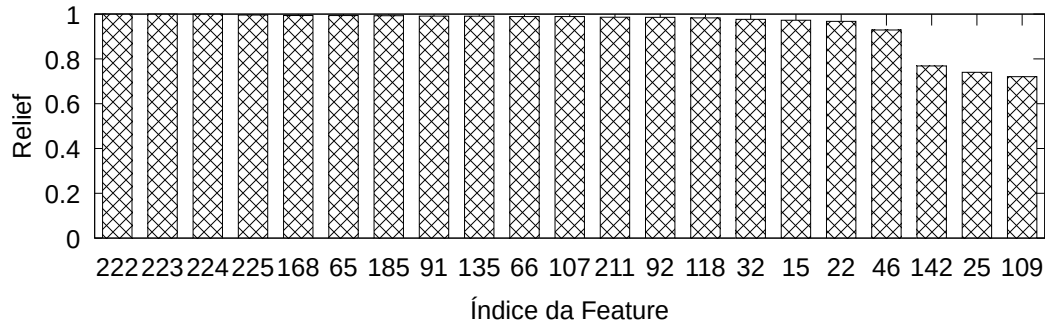


Figura 2. Top 20 selected features for WannaCry.

The *features* classified as the 20 most representative by the *relief* selection method are listed below: *CryptDecrypt* (F222), *CryptGenKey* (F223), *CryptExportKey* (F224), *CryptEncrypt* (F225), *SetFileAttributesW* (F168), *FindFirstFileExW* (F65), *MoveFileWithProgressW* (F185), *SetFileTime* (F91), *GetFileSizeEx* (F135), *NtQueryDirectoryFile* (F66), *CopyFileW* (F107), *CopyFileA* (F211), *DeleteFileW* (F92), *NtQueryInformationFile* (F118), *NtCreateFile* (F32), *CryptAcquireContextA* (F15), *GetFileAttributesW* (F22), *NtDuplicateObject* (F46), *RemoveDirectoryA* (F142), *NtEnumerateValueKey* (F25), *SHGetFolderPathW* (F109). The definitions of these can be found on the Microsoft Learn website³.

Most of the representative *features* for identifying the execution of the *WannaCry* ransomware consist of function calls from the *Windows* API (WinAPI). This API is a basic set of programming interfaces for the *Microsoft Windows* operating system.

The *CryptDecrypt* (F222) function was identified as the most representative. The *CryptDecrypt* API is a cryptography function from the *Windows* library used to decrypt encrypted data. Although it can be used to ensure data security, the *CryptDecrypt* API can also be used by *malwares* to decrypt data that was protected by security software or a regular user. This may allow the *malware* to access confidential information, such as passwords, credit card numbers, and other personal data.

Next, *CryptGenKey* (F223), *CryptExportKey* (F224), *CryptEncrypt* (F225), and *SetFileAttributesW* (F168) appear in positions 2 to 5 of the *Top 20*. The functions *CryptDecrypt*, *CryptGenKey*, *CryptExportKey*, and *CryptEncrypt*, although not necessarily malicious functions, are known to be used by *malwares*⁴. These are widely used functions in applications requiring data security and encryption. However, these functions can also be used by *malwares* to perform malicious activities, such as encrypting victims' data or creating new *threads* to carry out malicious operations in the background.

5.2. Ryuk Study

The *Ryuk* ransomware invokes several API calls to carry out its malicious activities. Some of these are used to encrypt system files, while others may be used to communicate with command servers, which in turn carry out the attack by encrypting the target's data. From Figure 3, it is possible to visualize the ranking of the *features* used by the *malware*. Below,

³Available at: <https://learn.microsoft.com/en-us/windows/win32/api/>

⁴Available at: <https://malapi.io/>

the selected *features* are listed, and the most relevant ones are discussed.

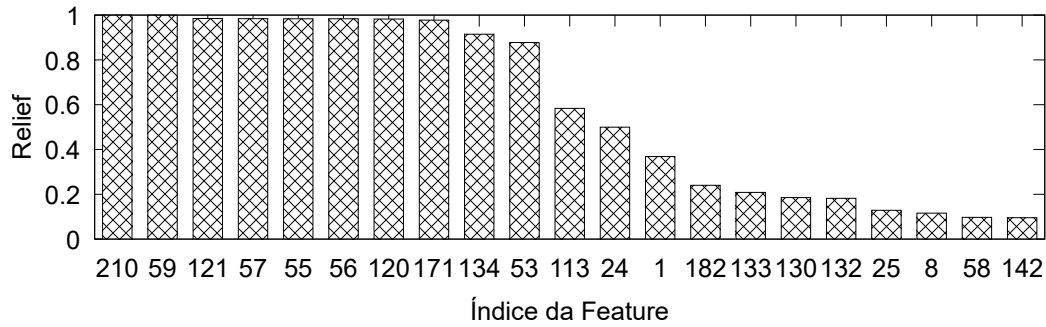


Figura 3. Top 20 selected features for Ryuk.

From the 211 *features* in the *dataset*, the Relief selection algorithm ranked the 20 most representative. They are: *EnumServicesStatusW* (F210), *LookupAccountSidW* (F59), *NtOpenKeyEx* (F121), *Process32NextW* (F57), *CreateToolhelp32Snapshot* (F55), *Process32FirstW* (F56), *NtQueryKey* (F120), *ShellExecuteExW* (F171), *GetVolumePathNamesForVolumeNameW* (F134), *GlobalMemoryStatusEx* (F53), *NtTerminateProcess* (F113), *NtQueryValueKey* (F24), *score* (F1), *LookupPrivilegeValueW* (F182), *RegEnumKeyW* (F133), *OleInitialize* (F130), *GetVolumeNameForVolumeMountPointW* (F132), *NtClose* (F25), *SetUnhandledExceptionFilter* (F8), *NtOpenProcess* (F58), *CreateProcessInternalW* (F142).

Below are the definitions of the main *features* identified: The *feature EnumServicesStatusW* (F210) enumerates running services on *Windows*, obtaining information including their name, status, and type. *Ryuk* uses *EnumServicesStatusW* (F210) to check if certain security services, such as popular antivirus solutions, are running on the target machine, avoiding detection. It may also use *Windows Defender* as the default solution. The *feature LookupAccountSidW* (F59) is used to retrieve the account name associated with a given user SID, allowing determination of the user logged into the target machine. The function *NtOpenKeyEx* (F121) is used to open *Windows* registry keys and modify their settings, allowing *Ryuk* to execute automatically at system startup. The function *Process32NextW* (F57) is used to identify specific processes that may interfere with the execution or operations of the *malware*.

5.3. CryptoLocker Study

CryptoLocker has a large scale and popularity. In Figure 4, the result of the feature selection based on the studies of this *ransomware* can be observed.

From the 211 *features* generated, those classified as the 20 most representative by applying the Relief algorithm are: *EnumServicesStatusW* (F210), *LookupAccountSidW* (F59), *Process32NextW* (F57), *NtOpenKeyEx* (F121), *Process32FirstW* (F56), *CreateToolhelp32Snapshot* (F55), *NtQueryKey* (F120), *ShellExecuteExW* (F171), *GetVolumePathNamesForVolumeNameW* (F134), *GlobalMemoryStatusEx* (F53), *score* (F1), *NtTerminateProcess* (F113), *NtQueryValueKey* (F24), *LookupPrivilegeValueW* (F182), *RegEnumKeyW* (F133), *OleInitialize* (F130), *GetVolumeNameForVolumeMountPointW* (F132), *SetUnhandledExceptionFilter* (F8), *NtOpenProcess* (F58), *NtClose* (F25), *CreateProcessInternalW* (F142).

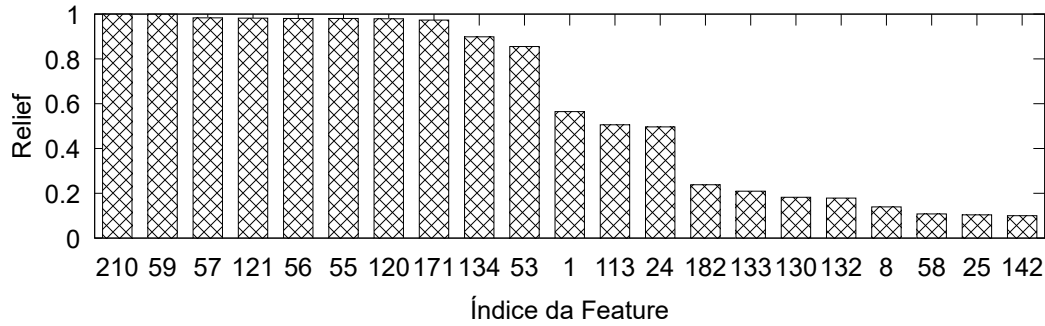


Figura 4. Top 20 selected features for CryptoLocker.

During its execution, the *malware* uses several APIs, with its main function being *EnumServicesStatusW* (F210). As mentioned earlier, this is the same function used by *malwares* to enumerate running services on *Windows*. The other three functions used by the *malware* are: the *LookupAccountSidW* (F59) function, which allows the *malware* to retrieve the name of the account associated with a given user SID (Security Identifier); the *Process32NextW* (F57) function, which, as previously mentioned, is used to list processes running on the system and allows the *malware* to identify and disable services; and finally, the *NtOpenKeyEx* (F121) function, which is used to open a specific key in the *Windows* registry.

O *ransomware CryptoLocker* a utiliza para abrir a chave do registro “Run” e adicionar um valor que aponta para seu próprio arquivo de execução, garantindo que o *malware* seja executado automaticamente sempre que o sistema é iniciado. Essa técnica é conhecida como “persistência de inicialização” e é usada por muitos tipos de *malwares* para garantir sua execução contínua. É importante monitorar as chaves críticas do registro para detectar e impedir tentativas de modificação maliciosa por parte de *malwares*.

6. Conclusion and Future Works

This work presented a study on feature selection for detecting the *Ransomwares* WannaCry, Ryuk, and CryptoLocker. It is important to highlight the relevance of the study presented in this work for the understanding of *malwares* and for the development of new techniques for preventing their attacks. These *ransomwares* are examples of cyber threats that can cause significant damage to organizations, ranging from the loss of data and valuable information to financial losses. Therefore, the study of these *malwares* is essential for improving cybersecurity and preventing potential damage. The study and understanding of each API mentioned in this research can assist in the creation of more effective security measures to combat *malware* attacks.

As future works, we intend to (i) use GRASP-FS [Quincozes et al. 2022] to refine the feature selection process, (ii) apply machine learning algorithms to evaluate performance in detection, (iii) study the behavior of *ransomwares* in other operating systems, and (iv) analyze the behavior of other *ransomwares* focused on infecting ICSs.

Referências

Abbasi, M. S., Al-Sahaf, H., Mansoori, M., and Welch, I. (2022). Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for

- feature selection. *Applied Soft Computing*, 121:108744.
- Abbasi, M. S., Al-Sahaf, H., and Welch, I. (2020). Particle Swarm Optimization: A Wrapper-Based Feature Selection Method for Ransomware Detection and Classification. In *Applications of Evolutionary Computation: 23rd European Conference, EvoApplications 2020*, pages 181–196. Springer.
- Beaman, C., Barkworth, A., Akande, T. D., Hakak, S., and Khan, M. K. (2021). Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & Security*, 111:102490.
- Brewer, R. (2016). Ransomware attacks: detection, prevention and cure. *Network security*, 2016(9):5–9.
- Chen, Q. and Bridges, R. A. (2017). Automated behavioral analysis of malware: A case study of wannacry ransomware. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 454–460.
- Chen, Q., Islam, S. R., Haswell, H., and Bridges, R. A. (2019). Automated ransomware behavior analysis: Pattern extraction and early detection. In *International Conference on Science of Cyber Security*, pages 199–214. Springer.
- CIS (2022). Top 10 Malware January 2022. <https://www.cisecurity.org/insights/blog/top-10-malware-january-2022>. Acesso em: Junho/2023.
- Curran, K. (2020). Cyber security and the remote workforce. *Computer Fraud & Security*, 2020(6):11–12.
- Damodaran, A., Troia, F. D., Visaggio, C. A., Austin, T. H., and Stamp, M. (2017). A comparison of static, dynamic, and hybrid analysis for malware detection. *Journal of Computer Virology and Hacking Techniques*, 13(1):1–12.
- Gera, T., Singh, J., Mehbodniya, A., Webber, J. L., Shabaz, M., and Thakur, D. (2021). Dominant feature selection and machine learning-based hybrid approach to analyze android ransomware. *Security and Communication Networks*, 2021.
- IBM (2022). O que é ransomware? Publicado em: <https://www.ibm.com/br-pt/topics/ransomware>. Acesso em: Junho de 2023.
- Ilić, S. Ž., Gnjatović, M. J., Popović, B. M., and Maček, N. D. (2022). A pilot comparative analysis of the cuckoo and drakvuf sandboxes: An end-user perspective. *Vojnotehnički glasnik*, 70(2):372–392.
- Kaspersky (2021). Kaspersky Security Bulletin 2021 Statistics. https://go.kaspersky.com/rs/802-IJN-240/images/KSB_statistics_2021_eng.pdf. Acesso em: Junho/2023.
- Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In Sleeman, D. H. and Edwards, P., editors, *Ninth International Workshop on Machine Learning*, pages 249–256. Morgan Kaufmann.
- Masid, A. G., Higuera, J. B., Higuera, J.-R. B., and Montalvo, J. A. S. (2022). Application of the sama methodology to ryuk malware. *Journal of Computer Virology and Hacking Techniques*, pages 1–34.

- Megchelenbrink, W. (2010). Relief-based feature selection in bioinformatics: detecting functional specificity residues from multiple sequence alignments. Master's thesis, Radboud University Nijmegen, the Netherlands.
- Micro, T. (2020). 2020 Report on Threats Affecting ICS Endpoints. Disponível em: https://documents.trendmicro.com/assets/white_papers/wp-2020-report-on-threats-affecting-critical-industrial-endpoints.pdf. Acesso em: Junho/2023.
- Micro, T. (2022). Defending the expanding attack surface. Disponível em: <https://documents.trendmicro.com/assets/rpt/rpt-defending-the-expanding-attack-surface-trend-micro-2022-midyear-cybersecurity-report.pdf>. Acesso em: Junho/2023.
- Oktavianto, D. and Muhandianto, I. (2013). *Cuckoo malware analysis*. Packt Publishing.
- Quincozes, S. E., Kazienko, J. F., and Copetti, A. (2018). Avaliação de conjuntos de atributos para a detecção de ataques de personificação na internet das coisas. In *VIII Simpósio Brasileiro de Engenharia de Sistemas Computacionais*, Porto Alegre. SBC.
- Quincozes, S. E., Mossé, D., Passos, D., Albuquerque, C., Ochi, L. S., and dos Santos, V. F. (2022). On the Performance of GRASP-Based Feature Selection for CPS Intrusion Detection. *IEEE Transactions on Netw. and Service Management*, 19(1):614–626.
- Razaulla, S., Fachkha, C., Markarian, C., Gawanmeh, A., Mansoor, W., Fung, B. C., and Assi, C. (2023). The age of ransomware: A survey on the evolution, taxonomy, and research directions. *IEEE Access*.
- Uppal, D., Mehra, V., and Verma, V. (2014). Basic survey on malware analysis, tools and techniques. *Int. J. Jnl on Computational Sciences & Applications (IJCSA)*, 4(1):103.
- Urooj, U., Al-rimy, B. A. S., Zainal, A., Ghaleb, F. A., and Rassam, M. A. (2022). Ransomware detection using the dynamic analysis and machine learning: A survey and research directions. *Applied Sciences*, 12(1):172.